Watermarking is not cryptography

Ingemar Cox, Gwenaël Doërr, Teddy Furon

▶ **To cite this version:**

Ingemar Cox, Gwenaël Doërr, Teddy Furon. Watermarking is not cryptography. Proc. Int. Work. on Digital Watermarking, inivited talk, 2006, Jeju island, South Korea. inria-00504528

HAL Id: inria-00504528
https://hal.inria.fr/inria-00504528

Submitted on 29 Mar 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Watermarking Is Not Cryptography

Ingemar J. Cox[1], Gwenaël Doërr[1], and Teddy Furon[2]

[1] University College London
Adastral Park, Ross Building 2
Martlesham IP5 3RE, United Kingdom
{i.cox, g.doerr}@adastral.ucl.ac.uk
http://www.adastral.ucl.ac.uk
[2] INRIA / TEMICS
Campus Universitaire de Beaulieu
35042 Rennes Cedex, France
teddy.furon@irisa.fr
http://www.irisa.fr

Abstract. A number of analogies to cryptographic concepts have been made about watermarking. In this paper, we argue that these analogies are misleading or incorrect, and highlight several analogies to support our argument. We believe that the fundamental role of watermarking is the reliable embedding and detection of information and should therefore be considered a form of communications. We note that the fields of communications and cryptography are quite distinct and while communications *systems* often combine technologies from the two fields, a layered architecture is applied that requires no knowledge of the layers above. We discuss how this layered approach can be applied to watermarking applications.

1 Introduction

Digital watermarking has received considerable attention as a complement to cryptography for the protection of digital content such as music, video and images. Cryptography provides a means for secure delivery of content to the consumer. Legitimate consumers are explicitly or implicitly provided with a key to decrypt the content in order to view or listen to it. Unfortunately, not all legitimate consumers are trustworthy and an untrustworthy consumer may alter or copy the decrypted content in a manner that is not permitted by the content owner. However, cryptography provides no protection once the content is decrypted, which is required for human perception. Watermarking complements cryptography by embedding a message within the content. If properly designed, the message remains in the content after decryption and, more importantly, after digital-to-analog and analog-to-digital conversion. By so doing, watermarking can be used to close the 'analog hole'[1].

[1] Not only must the digital content be decrypted, but it must also be converted to an analog signal in order for a person to see or hear it. This gives rise to the 'analog hole', which refers to the fact that all digital protection is lost at the point of perception. And this analog signal may be re-digitized by an untrustworthy consumer in order to obtain an unprotected digital copy of the content.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 1–15, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Since the primary motivation for watermarking has been for security, numerous analogies have been made between watermarking and cryptography. In this paper, we argue that many of these analogies are for the moment misleading or incorrect. We argue that watermarking should only be viewed as a means for reliably embedding and decoding information hidden in a cover Work. As such, it is a communication system, often modeled as spread spectrum communications or communications with side information. A system incorporating watermarking may also use cryptography but we argue that, up to now, a layered model has been much more successful than intermingling the two concepts.

To support our argument, we first provide a brief introduction to key concepts in communications (Section 2) and cryptography (Section 3). We then discuss the security requirements associated with watermarking. Section 4 highlights a number of cryptographic analogies used within the watermarking community and discusses the weaknesses of these analogies. A contrario, Section 5 shows that the layered model offers much safer designs with the examples of watermarking-based content authentication and watermarking-based traitor tracing. The last section extends this discussion to signal processing other than watermarking.

2 Communications

Communications is concerned with *reliable* transmission of a message from Alice to Bob over an *unreliable* channel. A channel is considered unreliable if there is a finite probability that an error will occur between the points of transmission and reception, e.g. Alice sends a '0'-bit, but Bob decodes a '1'-bit. Reliable communications is concerned with bandwidth, power or signal-to-noise ratio (SNR), channel coding and bit error rate (BER).

It was, of course, Shannon [1] who showed that the maximum rate of error free transmission, i.e. the channel capacity (in bits per second), is given by:

$$C = 2B \log_2 \left(1 + \frac{s}{n}\right) \tag{1}$$

where B denotes bandwidth in hertz, and s and n the signal and white Gaussian noise powers respectively. In order to approach this limit, it is necessary to encode the message m, prior to transmission. This channel code provides a level of redundancy that is measured by the code rate, R. For example, if every k-bits of the message are represented by an n-bit code, then the rate is $R = k/n$, where $n > k$. Finally the BER is a direct measure of the error rate achieved by a particular code and is usually plotted as a function of the SNR.

The sources of bit errors are many. The most common error model is Gaussian noise, but there are many other error sources. However, all such sources are usually considered to be naturally occurring and not due to the effects of an adversary. In fact, it is very rare for a civilian communications system to consider a hostile channel. However, military communications must do so. In a hostile military environment, the two primary concerns are (i) jamming and (ii) detection. Jamming refers to attempts by an active adversary to prevent Bob receiving a signal. Detection refers to an adversary's efforts to detect (and localize)

enemy communications. If this is successfully achieved then military firepower may be used to destroy the communications. Note that at this level, the concern is with the delivery of bits, not with the security of the bits (which is discussed in the next section). Secure communication is irrelevant if Bob never receives the communications!

Spread spectrum (SS) communications was originally developed to protect military communications from detection and jamming [2], although it is now widely used in many civilian applications, e.g. mobile phones. The basic principle behind SS communications is that each message bit is multiplied by a (pseudo random) chip sequence that spreads the message bit over a much broader spectrum. For example, consider an implementation of SS communications based on frequency hopping. Here, the original message bit is transmitted as n lower power bits (the chip sequence), each of which is transmitted over a separate frequency band that is pseudo-randomly chosen. The receiver is synchronized with the transmitter and also has knowledge of the pseudo-random sequence of frequency bands being used. Thus, Bob is able to sum the lower energy in each of the individual bands to produce a good signal-to-noise ratio (SNR) at the receiver.

However, an adversary has much greater difficulty detecting the transmission, since Eve does not know the pseudo-random frequency hopping sequence. If Eve monitors just one frequency band, she cannot be confident that there is any communication, since the signal transmitted is very weak and only persists for a short time. Furthermore, Eve cannot jam the channel as a precaution against possible communications. This is because the power needed to confidently jam all the frequency channels would be impractically large.

Another communications model that has received recent interest is known as communications with side information. Here the channel has two noise sources, both of which are unknown to the receiver, but the first of which is entirely known to the transmitter. Under these circumstances, which arise in mobile telephony and digital watermarking, how much information can Alice reliably transmit to Bob? Costa [3] proved that the channel capacity is the same as if the first noise source is absent.

3 Cryptography

Cryptography is concerned with the *secure* transmission of a message from a sender, Alice, to a recipient, Bob, over an insecure channel. A channel is considered insecure if the bits sent by Alice may be read or altered by an adversary, Eve, prior to receipt by Bob. It is important to realize that an insecure channel is not an unreliable channel. In fact, cryptography often assumes reliable communications, i.e. Bob receives exactly the same bits sent by either Alice or Eve - there are no unintentional errors.

A secure transmission is concerned with (i) privacy, (ii) integrity and (iii) authentication. Privacy is concerned with ensuring that an adversary, Eve, can learn nothing about the message intended for Bob, by examining the encrypted

bits sent by Alice. Integrity is concerned with ensuring that Bob can be confident that the message has not been altered by Eve prior to receipt. And authentication is concerned with guaranteeing that the sender of the message is actually Alice and not an impostor.

To ensure privacy, cryptography assumes the existence of an encryption function , E(.), which takes a message, m, and a key, K, and outputs an encrypted message, c, i.e. $c = \mathrm{E}(m, K)$. It further assumes a decryption function, D(.) that takes an encrypted message, c and a key, K, and outputs a cleartext message, m, i.e. $m = \mathrm{D}(c, K) = \mathrm{D}(\mathrm{E}(m, K), K)$.

Shannon [4] defined perfect security as an encryption function in which an adversary, Eve, learns nothing about the message, m, by inspection of the ciphertext, c. Perfect security can be realized using a one-time pad. Unfortunately, a one-time pad is not practical in most situations. Consequently, modern cryptography is therefore concerned with the design of cryptographic algorithms which approximate perfect security while re-using a shared key, K. It is assumed that the encryption and decryption algorithms are known to all parties, including the adversary, Eve. This is known as Kerckhoffs' Principle [5] and reduces Eve's cryptanalysis problem to inferring the key, K.

If the length of the binary key is n-bits, the total number of keys is 2^n and is called the keyspace. For sufficiently large n, the keyspace is enormous and exhaustive enumeration or brute force search is infeasible. Note that cryptography assumes that Eve learns nothing about the true key, K, by trying a key, K', that is close to K in the sense of say Hamming distance. In other words, if Alice encodes a message twice, once using key, K and once using a key, K', that differs by only one bit from K, then the two encrypted ciphertexts will be completely different with no correlation between them. In reality, modern cryptographic algorithms only approximate these assumptions.

Cryptographic systems in which the encryption and decryption algorithm share the same key are known as symmetric key or private key systems. One problem with such is how to initiate the system, i.e. how do Alice and Bob agree on a key without sharing this knowledge with Eve? Public key or asymmetric key cryptography solves this problem by assigning two keys to each individual: a public one (PK) that is published on a database and a secret one (SK) which is never disclosed. Everybody knows the public key of everybody. The main feature of public key watermarking lies in the asymmetry of the keys used during encryption and decryption, namely $m = \mathrm{D}(\mathrm{E}(m, PK), SK)$. For instance, Alice can encrypt the message she wishes to transmit with Bob's public key (PK_B). The resulting ciphertext $c = \mathrm{E}(m, PK_B)$ can then only be decrypted with Bob's secret key (SK_B) i.e. by Bob himself. In other words, the message m has been sent securely without agreeing on a secret key beforehand[2].

Integrity is guaranteed through the use of another cryptographic primitive known as a one-way hash function. This is a function that takes an arbitrarily

[2] However, for practical reasons, public key cryptography is usually used to exchange a key at the beginning of a transmission. The subsequent messages are then encrypted/decrypted with a private key crypto-system using the agreed session key.

long bit sequence (the pre-image) and outputs a constant length bit sequence known as a hash or digest. The characteristics of a hash function are:

1. it is easy to compute the hash value given a pre-image,
2. it is computationally unfeasible to generate a pre-image that hashes to a particular value,
3. it is hard to generate two pre-images with the same hash value, and
4. a single bit change in the pre-image results in a major change of the hash value.

The properties of a hash make it well-suited for guaranteeing the integrity of a message and the authenticity of its sender. For instance, Alice computes a hash of her message concatenated with the shared secret key, K, and appends the hash to the end of the message. This is one way to make what cryptographers call a message authentication code (MAC). Note that the message need not be encrypted if privacy is not an issue. On receipt, Bob can take the received message and compute the hash of the received message concatenated with their shared secret key. If this recomputed hash is identical to the hash appended by Alice, then Bob can be confident that the message has not been tampered with. While Eve may alter the unencrypted message, she is unable to compute the associated hash since Eve does not know the key shared by Alice and Bob. Consequently, any alteration made by Eve will be detected by Bob. Digital signatures combine hashing and public key encryption to guarantee a better authenticity and non repudiation while easing the key management.

4 Digital Watermarking

The most basic requirement of a digital watermarking system is the ability to embed and decode a message hidden within a cover Work. Applications of digital watermarking may require very much more. However, all systems need a reliable mechanism for embedding and decoding message bits[3]. We therefore believe that digital watermarking is fundamentally a form of communications and, as such, is primarily concerned with the reliable transmission of a message over an unreliable channel.

Of course, applications of digital watermarking also have security concerns. Security threats depend on the watermark application. However, the categories of attacks that have been identified are:

1. Unauthorized embedding,
2. Unauthorized decoding, and
3. Unauthorized removal.

Since much of the motivation for watermarking is driven by security concerns, it is not surprising that analogies have been made between watermarking and

[3] Even fragile watermarks must provide a reliable communications channel in the absence of distortions.

cryptography. However, while there are superficial similarities, we believe that most of these analogies are flawed. As an example, let us consider keyspaces in watermarking and cryptography.

4.1 The Keyspace Analogy

Some articles have studied the security of watermarking schemes with information theoretical tools [6,7,8], and especially equivocation. Here we rephrase this analysis in a simpler manner, thanks to a keyspace analogy. This analogy assumes that a watermarking technique is reliable because it is keyed by a n-sample sequence just like a crypto-system keyed by a n-bit secret. This analysis is not generic, we only consider the example of SS.

The security of a crypto-system is usually assessed by a common feature: the keyspace. The key θ randomly chosen from a keyspace Θ is usually a binary string made of n bits. An adversary without any *a priori* knowledge of the secret key can simply exhaustively test all the elements of the keyspace. This strategy is referred to as a brute force attack. The size of the keyspace is $|\Theta| = 2^n$. For each tested element, the probability P that it is equal to secret key is $P = 2^{-n}$ or $\log_2(P) = -n$. In other words, the larger the number of bits in the key, the lower is this probability P and the more time will be required to disclose the secret θ. For large n, say $n = 256$, the probability is negligibly small.

The concept of a key has been adopted by the watermarking community. For example, in spread spectrum watermarking, the key is used as a seed to a pseudo-random number generator that creates a binary antipodal sequence used as a carrier or chip sequence. Alternatively, the key may directly refer to the chip sequence. Nevertheless the behaviour of these two keyspaces is very different!

First, the 2^n possible antipodal binary sequences are not all eligible to serve as spread spectrum chip sequences. For instance, zero-average chip sequences are preferred to avoid affecting the direct component (DC) of the host signal, e.g. the average brightness of an image. This constraint reduces the number of possible keys to (in terms of bits):

$$\log_2 |\Theta| = \log_2 \binom{n}{n/2} \simeq n - \frac{1}{2}\log_2(n). \tag{2}$$

The approximation in (2) shows that despite this constraint, the size of the set is almost exponential and thus not drastically reduced.

However, there is a second, more serious difference with cryptography: the secret carrier does not need to be exactly disclosed in order to break the watermarking system. An attacker simply needs a close enough estimate! The more correlated the attacker's estimate is to the true chip sequence (i.e. secret key), the less distortion is required to remove the watermark. Indeed, practical studies have shown that a normalised correlation greater or equal to $\rho_{\min} = 0.4$ between the attacker's estimate and the true key, is sufficient to remove a watermark while maintaining good perceptual quality [6].

In other words, if at least $k_{\min} = \lceil n(\rho_{\min} + 1)/2 \rceil$ samples of the estimated carrier match the ones of the secret carrier, the attack will be successful. Keeping

in mind that half these 'matching samples' need to be 1's to preserve the zero average, the probability P that a randomly picked eligible carrier leads to a successful attack is given by:

$$P = \sum_{\substack{k_{\min} \le k \le n \\ k \text{ even}}} \binom{n/2}{k/2}^2 \Bigg/ \binom{n}{n/2}. \tag{3}$$

Numerical computations show that $\log_2(P) \approx -0.12\,n$ bits for $\rho_{\min} = 0.4$.

This is a remarkable difference! In simple terms, the cryptanalyst looks for the one and only unique secret key among 2^n eligible elements, whereas the watermark hacker looks for one of the $2^{0.88n}/\sqrt{n}$ suitable carriers among a set of $2^n/\sqrt{n}$, i.e. the search space is only $2^{0.12n}$.

Furthermore, we note that it has been proven that information about the secret key leaks from watermarked content (at least for spread spectrum [6] and lattice quantization index modulation [9] watermarking schemes). Thus, observations of watermarked content give the pirate strong *a priori* knowledge with which to estimate the key.

The keyspace analogy shows that the belief (a n-sample watermarking carrier provides as much security as a n-bit cryptographic key) is clearly flawed and highly misleading.

4.2 The Public Key Analogy

As discussed in Section 3, public key cryptography provides a mechanism for Alice and Bob to initiate a secure communication without first having to share a secret key. In watermarking, we would like to permit an untrustworthy third party, Eve, to read a watermark embedded in a Work. However, if we grant this capability, we do not want Eve to be able to remove the watermark from the content. The capability to read-but-not-remove, is almost a holy grail of digital watermarking.

Unfortunately, as of the time of writing, there is neither a theoretical proof on the feasibility of this capability, nor a practical watermark algorithm, that we are aware of.

Indeed, a number of papers have been published [10,11,12] that seek to create read-but-not-remove watermarking systems based on an analogy to public key cryptography. The analogy is that the embedder will embed the watermark with one key, and the detector will detect the watermark with another. Since these two keys are different, perhaps this will prevent an adversary with a detector from removing the watermark.

This analogy is not just flawed, it is wrong [13]. It is true that these schemes provide a better robustness against average attack, Principal Component Analysis (PCA), and oracle attack. However, the disclosure of the detection key permits specialized closest-point attacks that prevent detection while maintaining a good perceptual quality [14]. Hence, they are bad candidates for providing the read-but-not-remove capability. This proves that asymmetry is not sufficient, and

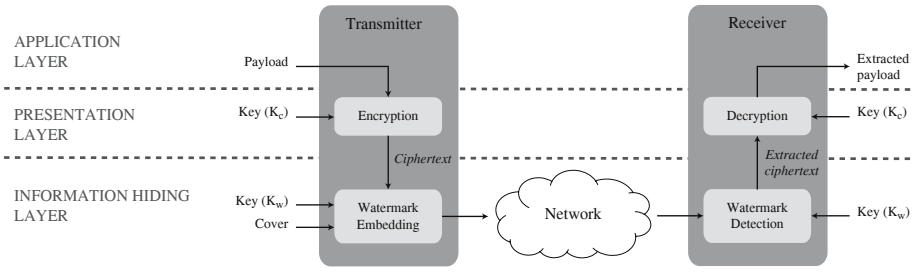

Fig. 1. Layered architecture for watermarking systems: a cryptographic primitive is simply added on top of the watermarking algorithm to provide security

indeed, it is not sure that it is even necessary to achieve the read-but-not-remove capability [15].

5 The Layered Approach

We believe that a layered approach to the design of secure watermarking systems, rather than an intermingling of the two fields of watermarking and cryptography, better ensures security for most applications.

We call a layered architecture a system where cryptography and watermarking primitives are well separated, as depicted in Figure 1, to implement a complete watermarking *system*. This approach is first motivated by its similarity with the popular open systems interconnection model (OSI model [16]). Using the OSI terminology, the information hiding layer is represented by the session layer (synchronization), the transport layer (error correction) and the physical layer (transmission). The presentation layer (encryption) is overlaid on top and the cleartext can be retrieved at the highest layer. Indeed, the 'encryption' box in Figure 1 has to be understood in a wide sense: the watermarking algorithm receives as input, the output of any cryptographic primitive, not only encryption.

These two layers have very different levels of security. The lower level, watermarking, has no security in the cryptographic sense. That is, it provides no protection with respect to privacy, authentication or integrity. It is only concerned with the reliably transmitting the (encrypted) bits. Since security for watermarking encompasses not just cryptographic security, (privacy, authentication, integrity), but also detection and removal, it is almost always the weakest link, as illustrated in Section 4.1 by the keyspace analogy.

Assume that cryptography is infinitely more secure than watermarking. Hiding a ciphertext, as depicted in Figure 1, forbids unauthorized embedding and decoding. However, it is absolutely useless against watermark removal or jamming. Hence, the layered approach does not necessarily bring a higher security level. However, it clearly separates the functionality of the two complementary technologies and reduces the risk of applying an inappropriate technique to solve a specific security issue.

5.1 Case Studies

Content Authentication. In this context, the goal is to ensure that the protected content has not been tampered with. Both cryptography and watermarking offer a technical solution to this challenge. Nevertheless, each one of them has its own shortcoming.

In cryptography, authentication is achieved by appending a digital signature or MAC to the content. However, 'appending' something to the content introduces an overhead i.e. more information has to be transmitted. And there is the risk that the MAC may be "lost" during format conversions.

Early proposals in watermarking suggested to embed a client-dependent watermark in the content. For example, the least significant bits (LSB) of an image are set to match a specific pseudo-random sequence. In this case, the drawback is that the watermark signal is not dependent of the protected content and can be copied to another one [17].

Combining both technologies immediately comes to mind as a means to avoid these shortcomings. One can indeed embed a watermark which encodes the digital signature or MAC of the content. When receiving a content, the user simply has to compare the digital signature of the content with the one stored by watermarking to validate the authenticity of the document. In this case, the watermark can no longer be copied from one content to the other because the watermark is now content-dependent. Moreover, no overhead is introduced. The only caution to be taken is that the watermarking process should not modify the bits of the content used to compute the digital signature of MAC. For instance, if the watermark is embedded in the LSB of an image, the digital signature should only be computed on the 7 most significant bits (MSB) of the pixels in the image.

A cryptographic hash provides exact authentication: a single bit change and the content is reported to be corrupted. In practice, a more flexible authentication of multimedia content may be desired to account for the various signal processing primitives (filtering, lossy compressions, etc) that do not modify the *semantic* meaning of the content. This has led to the introduction of 'robust hash functions' which will be further described in Section 6.1.

Traitor Tracing. In a typical fingerprinting scenario, Alice owns a few high valued multimedia items and wants to distribute these to a large number of customers. However, she is concerned that one or more of the customers may illegally redistribute her assets, thus inducing a loss of revenues. To address this issue, Alice introduces some customer-dependent modifications before distributing her assets. If a copy is found on an illegal distribution network, Alice looks for these modifications which serve as a fingerprint to trace back the 'traitor' who has broken his/her license agreement. Knowing Alice's strategy, a small set of users, usually referred to as a *collusion set*, may compare their individual copies, detect where they differ and create a new copy which potentially no longer contains a valid fingerprint. Alice's goal is then to design her system so that she can cope with such behaviours.

Traitor tracing has been studied by cryptographers where the problem is usually cast as the design of collusion-secure codes. A fundamental hypothesis is that, in combining their fingerprinted versions, a collusion inherently obeys a rule, called the marking assumption [18]. The most common marking assumption is that the set of colluders can only alter those bits of the codeword that differ between colluders. That is, those bits of each colluder's fingerprint (codeword) that are identical for all the colluders remain unchanged after a collusion attack. This assumption leads to the notion of a feasible set, also referred to as set of descendants, which is the collection of fingerprints that the collusion may produce. Traitor tracing needs collusion-secure codes designed so that each element in the feasible set can be linked back to at least one of the colluders as long as the number of colluders does not exceed a given limit.

Many recent fingerprinting solutions follow a layered architecture [19]. The cryptographic customer-dependent collusion-secure codeword is the payload embedded by the watermarking algorithm, and digital watermarking is simply exploited as a means to transmit the customer fingerprint, from one point to the other.

5.2 Knowledge of Lower Layers

In a layered model, the layers below do not need to know about the layers above. A function at layer i will accept inputs from layers above, but the function does not need to know how the inputs or outputs are interpreted by the layers above. However, the design and implementation of layers above may need a knowledge and understanding of the lower level protocols.

To illustrate this, let us re-examine the problem of traitor tracing[4]. The marking assumption on which collusion-secure codes are based is a model of the errors that can occur once the fingerprint is transmitted. These errors occur within the lower layers and can therefore be considered as a model of the 'noise' present in these levels. The most common marking assumption assumes that the set of colluders cannot alter those bits of the codewords that are common across all colluders.

We believe that this marking assumption is valid for watermarking algorithms that embed each bit of the fingerprint independently. For example, standard spread spectrum (SS) and quantization index modulation (QIM) techniques fit this model very well. However, more recent watermarking algorithms introduce dependency between successive embedded symbols in order to achieve higher embedding rates [23,24]. Thus, if one bit it altered, this may result in multiple successive bit errors at the decoder. Under these circumstances, the marking assumption would no longer be valid. For example, if two colluders are assigned two fingerprints that differ in only one bit, then the marking assumption states that all the remaining bits should be preserved. However, this one bit error may introduce a burst error at the decoder. Therefore, the extracted codeword may differ in bits that are common to both colluders.

[4] We will assume that the information hiding layer is secure enough to avoid estimation attacks [20,6]. In other words, a proper key scheduling policy [21,22] is enforced to prevent information leakage and subsequent jamming of the watermarking channel.

This example highlights the need for the upper levels to understand the workings of the lower levels. If a lower level watermarking algorithm is based on SS or QIM, then traditional collusion codes are applicable. However, if the lower level watermarking algorithm is based on dirty paper trellis coding, then a different type of collusion code must be designed and used.

6 Extension to Other Signal Processing

We would like to end this paper by considering the relationships between cryptography and signal processing other than watermarking.

6.1 Robust Hash

Section 5.1 has highlighted the need for 'flexible' hash functions which would output the same binary hash for perceptually similar contents. The quest for such a functionality has been previously explored in multimedia indexing and biometrics. This is usually referred to as 'robust hash', perceptual hash, soft hash or passive fingerprint [25,26].

Ideally, a robust hash would have the properties of:

1. it is easy to compute the hash value given a pre-image,
2. it is computationally unfeasible to generate a pre-image that hashes to a particular value,
3. it is hard to generate two perceptually different pre-images with the same hash value, and
4. only a perceptually significant change to the pre-image results in a change to the hash value.

Only the last two properties are different from the definition of a hash provided in Section 3. Nevertheless, the notions of "perceptually different" and "perceptually significant change" are very difficult to define.

Many researchers have attempted to realize a robust hash by designing completely new 'hash' functions. However, the design of hash functions is very difficult as revealed by incremental works [27,28]. Even cryptographic hash functions such as MD3 and SHA-1 have recently been shown to be partially flawed.

A layered approach would continue to use a cryptographic hash. The robust hash would be built on top of the cryptographic hash in a manner proposed by [29]. In such a design, the robust hash accepts the input content, e.g. an image, and extracts a robust *representation* of the content. It is this robust representation that is crytographically hashed. And it is this robust representation that is only altered if the input content is perceptually altered. The advantage of this layered solution is obvious. First, we can utilized well-know and trusted cryptographic tools. And second, we can utilize the considerable body of work regarding robust representations of signals. Finally, if the robust hash fails, we know it must be a failure of the robust representation. However, without a layered approach, errors are more likely and their causes more difficult to determine.

6.2 Signal Processing in the Encrypted Domain

Traditionally signal processing has been applied prior to encryption. In fact, for many encryption algorithms, it would make no sense to apply signal processing to the encrypted signal. The result would be nonsense. However, there is recent interest in developing encryption algorithms that permit signal processing of the encrypted signal. The motivation stems from the need to perform signal processing operations on machines that may not be trusted. For example, when streaming encrypted content over the Web, proxy servers may need to perform transcoding in order to reduce the bandwidth of the signal to match the recipient's (fluctuating) channel capacity. However, the proxy server is not trusted by the content owner who therefore does not wish the proxy server to decrypt the signal prior to transcoding[5].

We do not believe that this paradigm breaks the layered model. Rather, the processing of the encrypted signal should occur above the encryption layer. At the signal processing layer, there is a need to understand the nature of the encryption algorithm in order to design properly functioning algorithms. However, from the lower level encryption/decryption perspective, it is irrelevant what signal processing has occurred between encryption and decryption, provided the resulting signal can be correctly decrypted.

7 Conclusion

The interest in digital watermarking is strongly motivated by multimedia security issues. Consequently, it is not surprising that a number of cryptographic analogies have been applied to watermarking. However, we have argued that many of these cryptographic analogies are misleading or incorrect.

To support our argument we examined the concept of keys used in both cryptography and watermarking and showed that their properties are very different. In particular, for spread spectrum watermarking, an n-bit key has a keyspace of only $2^{0.12n}$ which is very much less than the equivalent keyspace in cryptography. We also discussed the concept of public key watermarking and argued that this concept, i.e. read-but-not-write, does not arise from using different keys for embedding and detection.

Fundamentally, watermarking is communications and is therefore concerned with the reliable delivery of bits over an unreliable channel. This is not a problem that is addressed by cryptography. However, cryptography does have a role to play in the development of applications of watermarking. Specifically, well-known cryptographic algorithms can be used to guarantee the privacy, authenticity and integrity of messages embedded in multimedia content. However watermark security must also consider the threat of unauthorized removal, for which there is no cryptographic solution.

[5] This problem has been considered in [30]. However, their proposed solution does not require signal processing of the encrypted stream. Rather, the stream is split into several layers that are independently encrypted. Then, if transcoding is required, the high-resolution stream can simply be deleted.

As in traditional communication systems, we recommend the use of a layered architecture. In such a design, watermarking is responsible for the synchronization and delivery of bits while cryptography is responsible for guaranteeing their privacy, integrity and authenticity. This separation simplifies analysis and modification of application systems.

To demonstrate this, we discussed two application areas: content authentication and traitor tracing. We explained how a layered design using cryptography and watermarking can be used to provide for both exact and approximate authentication (robust hash). Our discussion of traitor tracing served to highlight the point that in a layered architecture it may be necessary for the higher layers to understand the details of the lower levels. However, the lower level functions do not need to know how the upper layers will interpret signals.

We briefly commented on the recent interest in applying signal processing to encrypted signals and observed that such an approach can still be accommodated within a layered framework.

In summary, we hope this paper has clearly distinguished the roles of digital watermarking and cryptography and encouraged the use of a layered framework to the design of watermarking applications. We hope that these considerations will be taken into account in future proposals to combine cryptograpy and watermarking, together with other rules of good design [31].

References

1. Shannon, C.: A mathematical theory of communication. Bell System Technical Journal **27** (1948) 379–423 & 623–656
2. Kahn, D.: Cryptology and the origins of spread spectrum. IEEE Spectrum **21** (1984) 70–80
3. Costa, M.: Writing on dirty paper. IEEE Transactions on Information Theory **29** (1983) 439–441
4. Shannon, C.: Communication theory of secrecy systems. Bell System Technical Journal **28** (1949) 656–715
5. Kerckhoffs, A.: La cryptographie militaire. Journal des sciences militaires **IX** (1883) 5–83
6. Cayre, F., Fontaine, C., Furon, T.: Watermarking security: Theory and practice. IEEE Transactions on Signal Processing, Supplement on Secure Media **53** (2005) 3976–3987
7. Comesaña, P., Pérez-Freire, L., Pérez-Gonzàlez, F.: Fundamentals of data hiding security and their applications to spread spectrum analysis. In: Proceedings of the 7th International Workshop on Information Hiding. Volume 3727 of LNCS. (2005) 146–160
8. Pérez-Freire, L., Comesaña, P., Pérez-Gonzàlez, F.: Information-theoretic analysis of security in side-informed data hiding. In: Proceedings of the 7th International Workshop on Information Hiding. Volume 3727 of LNCS. (2005) 131–145
9. Pérez-Freire, L., Pérez-Gonzàlez, F., Comesaña, P.: Secret dither estimation in lattice-quantization data hiding: a set-membership approach. In: Security, Steganography, and Watermarking of Multimedia Contents VIII. Volume 6072 of Proceedings of SPIE. (2006) 6072–0W

10. Hartung, F., Girod, B.: Fast public-key watermarking of compressed video. In: IEEE International Conference on Image Processing. Volume I. (1997) 528–531
11. Furon, T., Duhamel, P.: An asymmetric public detection watermarking technique. In: Proceedings of the Third Information Hiding Workshop. Volume 1768 of Lecture Notes in Computer Science. (1999) 88–100
12. Eggers, J., Su, J., Girod, B.: Public key watermarking by eigenvectors of linear transforms. In: Proceedings of the European Signal Processing Conference. Volume III. (2000)
13. Furon, T., Duhamel, P.: An asymmetric watermarking method. IEEE Transactions on Signal Processing, Special Issue on Signal Processing for Data Hiding in Digital Media and Secure Content Delivery **51** (2003) 981–995
14. Furon, T., Venturini, I., Duhamel, P.: Unified approach of asymmetric watermarking schemes. In: Security and Watermarking of Multimedia Contents III. Volume 4314 of Proceedings of SPIE. (2001) 269–279
15. Miller, M.L.: Is asymmetry watermarking necessary or sufficient? In: Proceedings of European Signal Processing Conference. Volume I. (2002) 292–294
16. Zimmermann, H.: OSI reference model - the ISO model of architecture for open systems interconnection. IEEE Transactions on Communications **28** (1980) 425–432
17. Kutter, M., Voloshynovskiy, S., Herrigel, A.: Watermark copy attack. In: Security and Watermarking of Multimedia Contents II. Volume 3971 of Proceedings of SPIE. (2000) 371–380
18. Boneh, D., Shaw, J.: Collusion secure fingerprinting for digital data. IEEE Transaction on Information Theory **44** (1998) 1897–1905
19. Trappe, W., Wu, M., Wang, Z., Liu, K.: Anti-collusion fingerprinting for multimedia. IEEE Transaction on Signal Processing, Special Issue on Signal Processing for Data Hiding in Digital Media and Secure Content Delivery **51** (2003) 1069–1087
20. Doërr, G., Dugelay, J.-L.: Security pitfalls of frame-by-frame approaches to video watermarking. IEEE Transactions on Signal Processing, Supplement on Secure Media **52** (2004) 2955–2964
21. Lin, E., Delp, E.: Temporal synchronization in video watermarking. IEEE Transactions on Signal Processing, Supplement on Secure Media **52** (2004) 3007–3022
22. Harmancı, Ö., Kucukgoz, M., Mıhçak, K.: Temporal synchronization of watermarked video using image hashing. In: Security, Steganography and Watermarking of Multimedia Contents VII. Volume 5681 of Proceedings of SPIE. (2005) 370–380
23. Miller, M.L., Doërr, G., Cox, I.J.: Applying informed coding and informed embedding to design a robust, high capacity watermark. IEEE Transactions on Image Processing **13** (2004) 792–807
24. Abrardo, A., Barni, M., Pérez-Gonzàlez, F., Mosquera, C.: Trellis-coded rational dither modulation for digital watermarking. In: Proceedings of the 4th International Workshop on Digital Watermarking. Volume 3710 of LNCS. (2005) 351–360
25. Fridrich, J., Goljan, M.: Robust hash functions for digital watermarking. In: Proceedings of the International Conference on Information Technology: Coding and Computing. (2000) 178–183
26. De Roover, C., De Vleeschouwer, C., Lefèbvre, F., Macq, B.: Robust video hashing based on radial projections of key frames. IEEE Transactions on Signal Processing, Supplement on Secure Media **53** (2005) 4020–4037
27. Kozat, S., Venkatesan, R., Mıhçak, K.: Robust perceptual image hashing via matrix invariants. In: Proceedings of the IEEE International Conference on Image Processing. Volume V. (2004) 3443–3446

28. Monga, V., Mıhçak, K.: Robust image hashing via non-negative matrix factorizations. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. Volume II. (2006) 225–228
29. Sun, Q., Ye, S., Lin, C.-Y., Chang, S.-F.: A crypto signature scheme for image authentication over wireless channel. International Journal of Image and Graphics **5** (2005) 1–14
30. Wee, S., Apostolopoulos, J.: Secure scalable streaming enabling transcoding without decryption. In: Proceedings of the IEEE International Conference on Image Processing. Volume 1. (2001) 437–440
31. Katzenbeisser, S.: On the integration of watermarks and cryptography. In: Proceedings of the 2nd International Workshop on Digital Watermarking. Volume 2939 of Lecture Notes in Computer Science. (2003) 50–60